

UM EECS 270 F22
Introduction to Logic Design

16. Sequential Timing Analysis

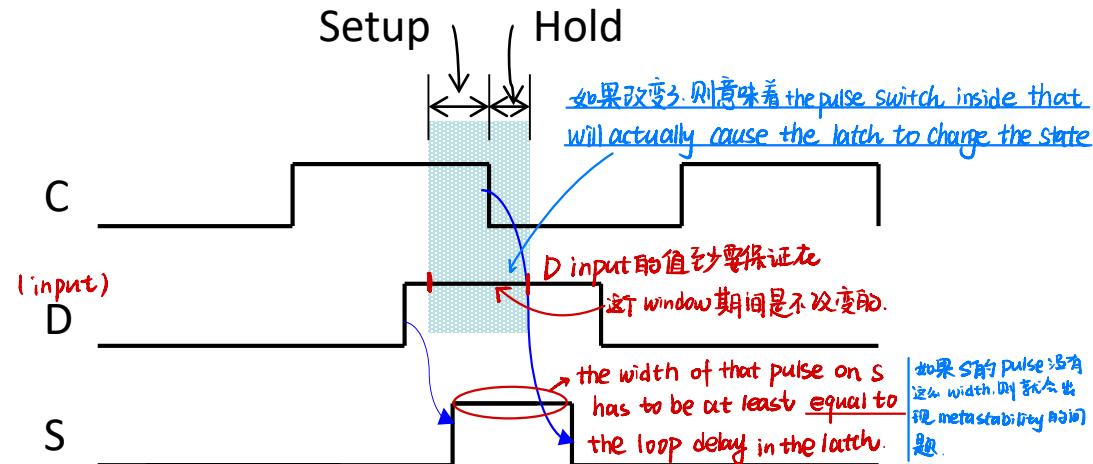
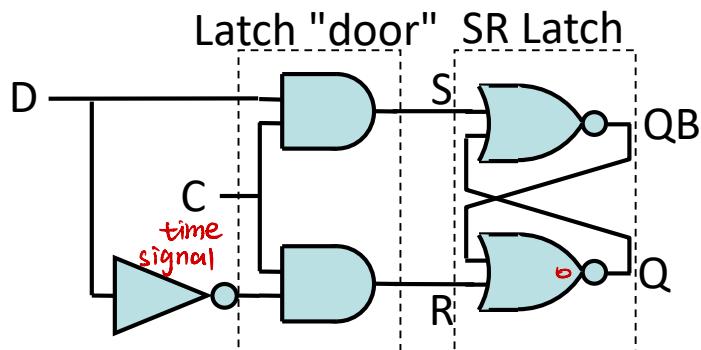
Timing Issues



- Setup and hold times/constraints
- Computing combinational delay • this is what we did before • But know we have to relate to the clock period. to figure out if combination of delays, both short and long delays (min/max) make sure that the clock period is sufficient long and pulse width is sufficient long for correct operation
- Flip-flop signal departure & arrival equations
- Computing minimum clock period at which the circuit might work.
- Clock skew meaning that you received the clock at slightly different time than somebody else
- Synchronizing external (asynchronous) inputs
- MTBF: Mean Time Between Failures

Setup and Hold Times

Positive Level-Sensitive D Latch



- To set, S must be held at 1 for at least 2 NOR delays
- 为了避免 metastability 的问题, 当我们要 flip Latch 的 State 时, 我们需要让 S 或 R state there at least the delay in the loop, 之后我们才能将值移除。不然, feedback action 没有足够的时问 to capture the new value.

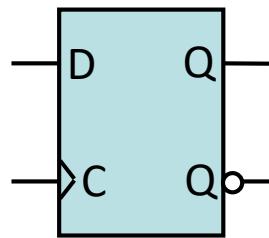
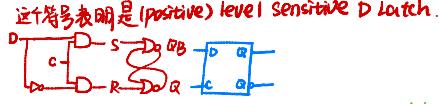
如果 SR 是 D, 那 feedback action is keeping the state $D \rightarrow QB, QB \rightarrow Q$. 若 S 为 1, 则 S 会改变 QB, →进而改变 Q. 等到 Q 也改变了, 则 Q 和 QB 异号之 to 0, 才可以 remove 'S'. 因此 feedback action will take over
- $S = D \& C$
- D must change to 1 at least 2 NOR delays before C goes to 0 (closes the latch)

Q. Figure out which edge of the clock actually closes the latch
→ means it disables the latch from seeing the input (Here is D)

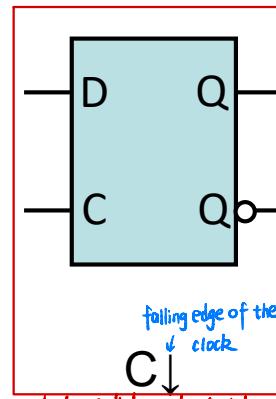
Setup and Hold Times

- By insuring that the D input is stable for a specified minimum length of time before (setup) and after (hold) the **appropriate clock edge** we eliminate metastability!
- Assume that setup and hold times are provided. They can be calculated, but the analysis is tricky.
- Which clock edge?

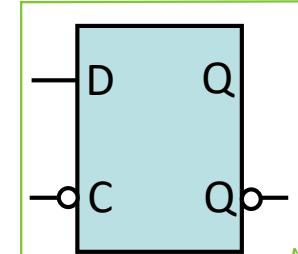
Edge that “closes” the latch



the leader latch is active on the low level of clock, so when the clock makes a rising transition, it separates



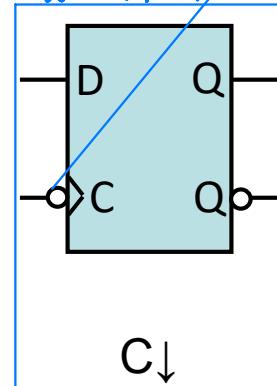
this one looks like what we did on the leader latch



it's active on the low level of the clock. So it's a negative level sensitive latch.

C↑

this is a negative edge triggered flip flop



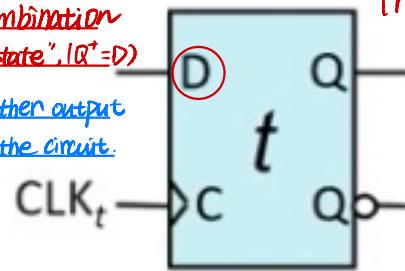
Setup and Hold Constraints

Positive Edge-Triggered D Flip-Flop

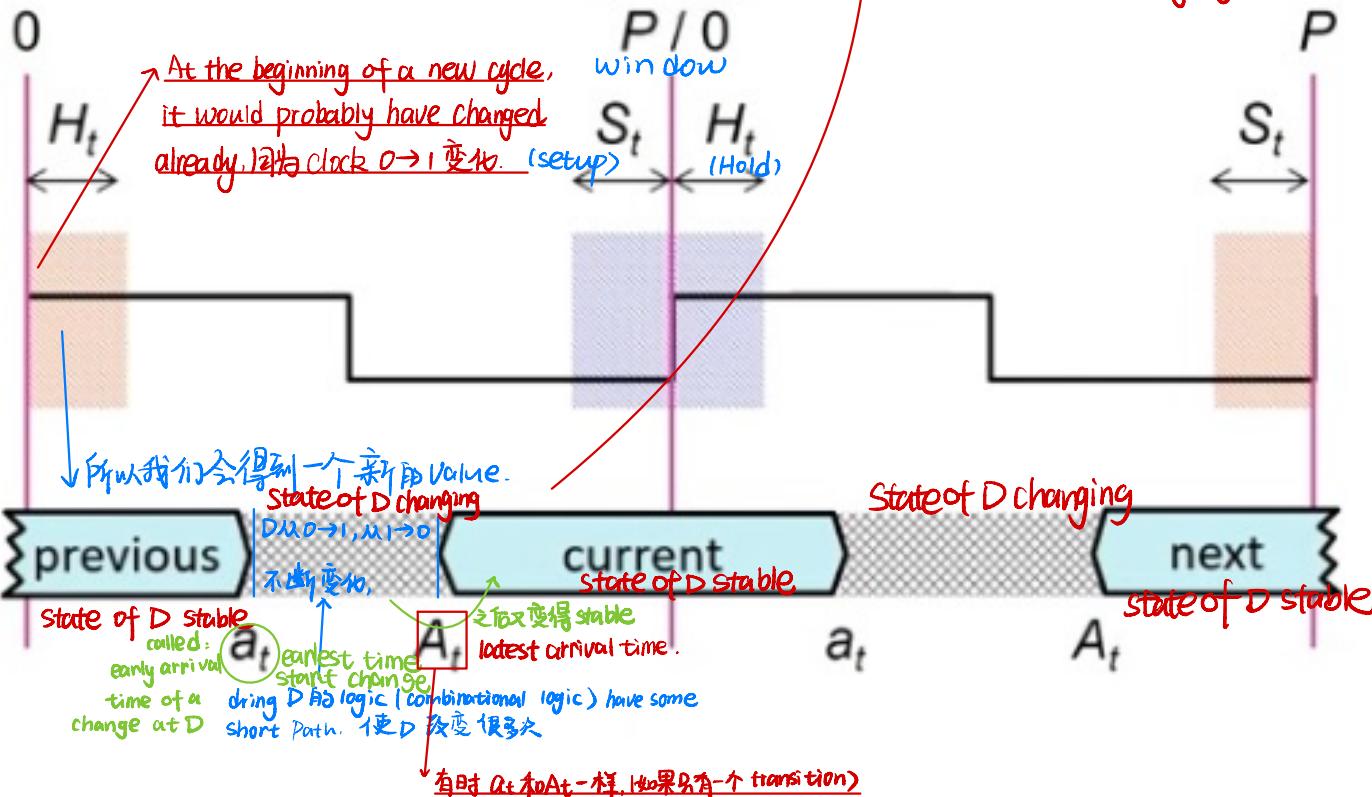
input D will be some combination of logic. D is the "next state". ($Q' = D$)

input $\overset{P}{\text{is}}$ Primary input to other output from other flip-flops in the circuit.

the closing edge of the clock is the rising edge.



Q: 我们想知道: When does it become stable, and when does it start changing.



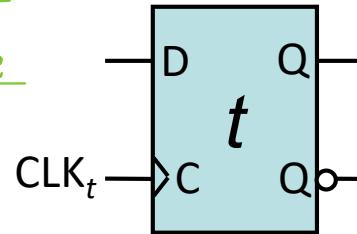
这个变化的过程是 repetitive 的, 所以我们可以忽略 driving this 的 combinational logic.

我们考虑 worst case, 即是 absolute worst case time that it will start changing at "at"

Setup and Hold Constraints

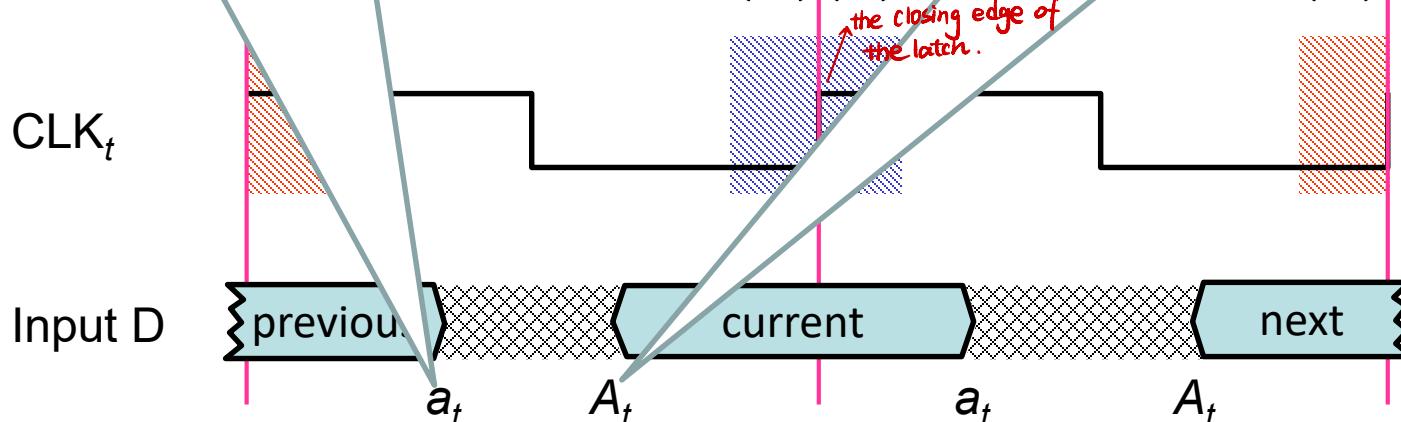
We are going to talk about a cycle,
 the cycle has a time scale from 0 to P.
 the next cycle will reset to 0 to P. and repetitive

Positive Edge-Triggered D Flip-Flop



Early "arrival" time.
 First signal transition in cycle

Late "arrival" time.
 Last signal transition in cycle



can not change sooner than
 the "hold time" of the latch.

Hold Requirement

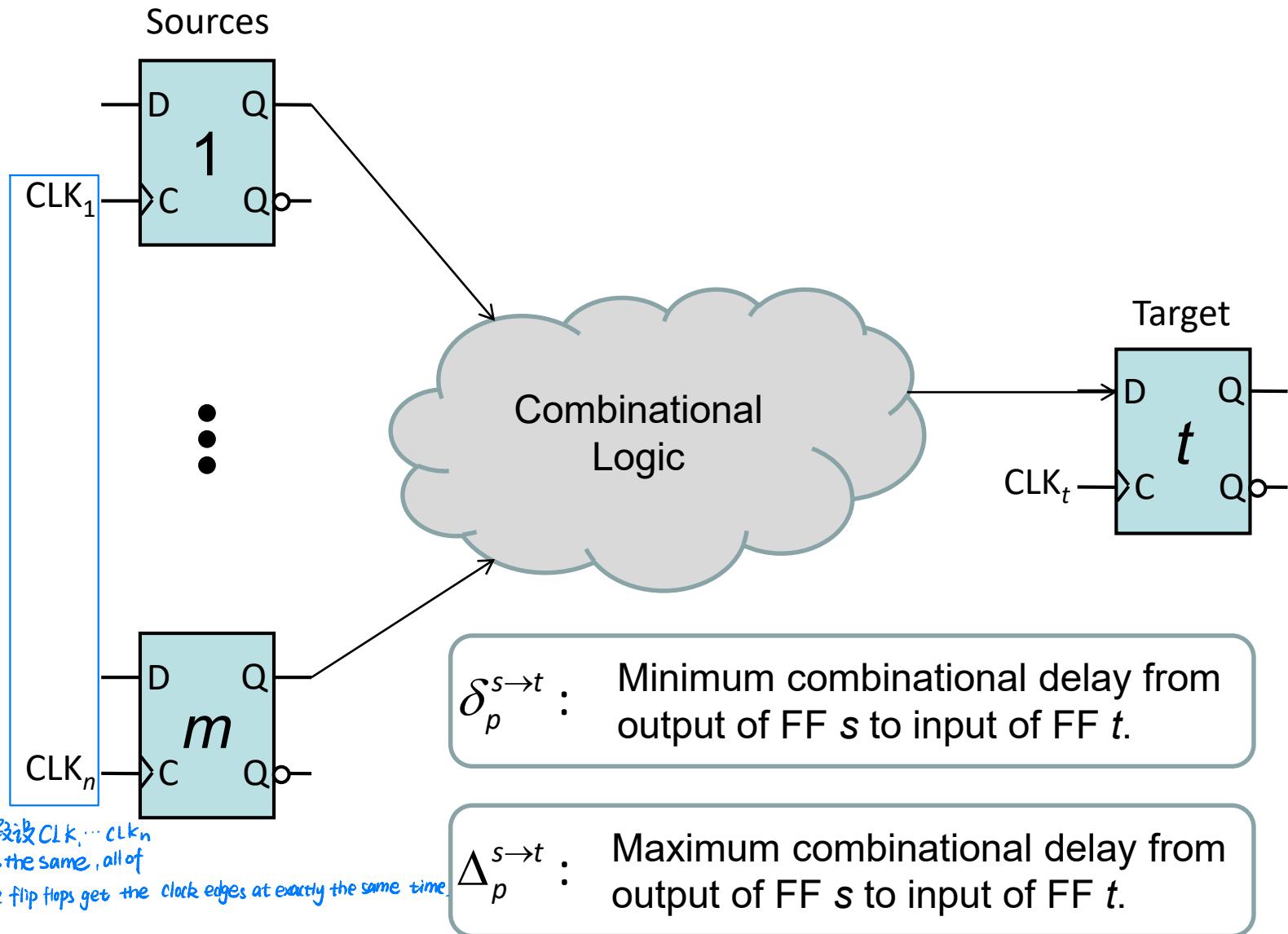
$$a_t \geq H_t$$

$$0 \leq a_t, A_t \leq P$$

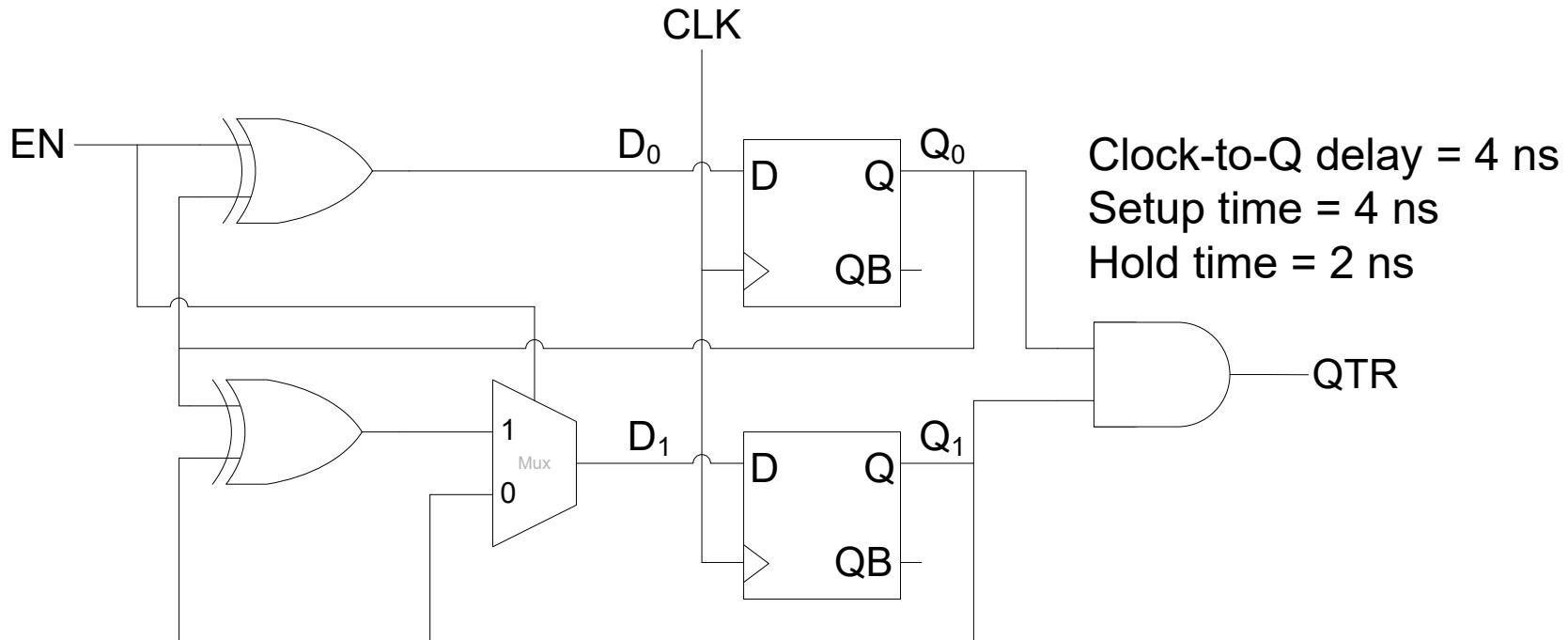
Setup Requirement

$$A_t \leq \underbrace{P - S_t}_{\text{period.}}$$

Combinational Delay Model



Timing Analysis Example



We need to compute the following minimum and maximum combinational propagation delays

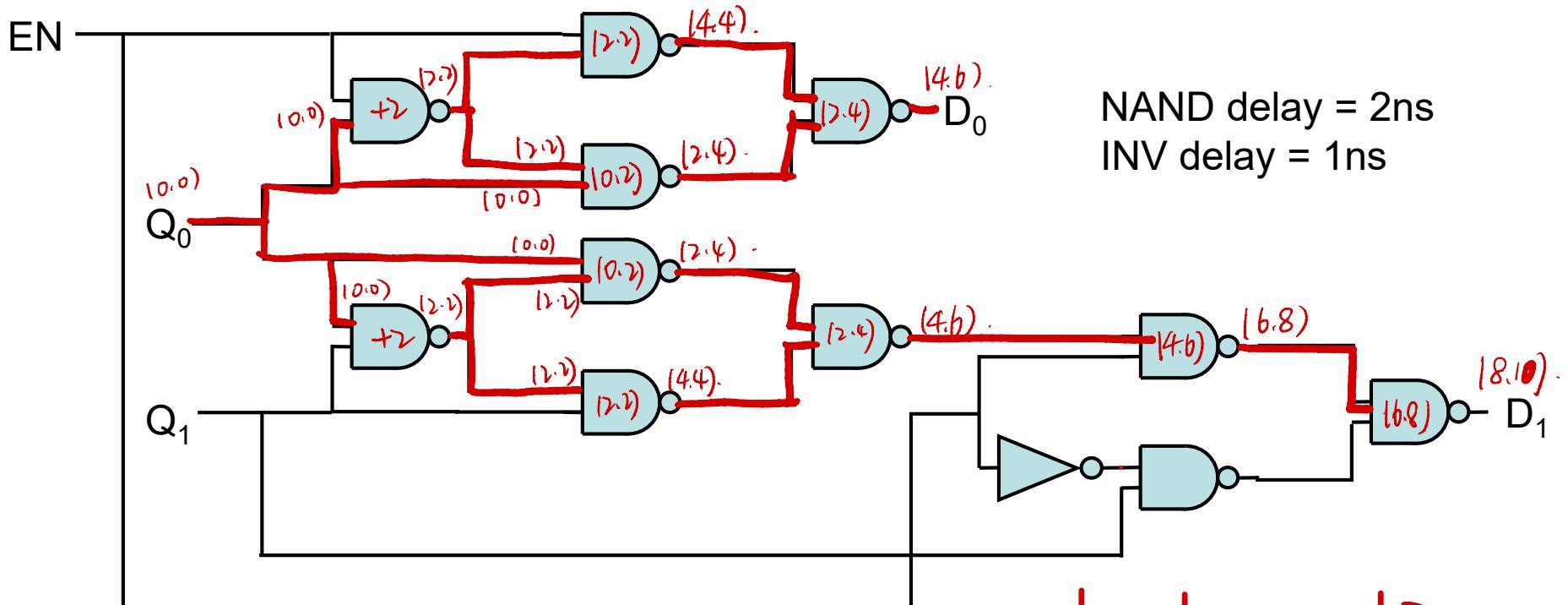
	D_0	D_1
EN	$[\delta_p^{EN \rightarrow D_0}, \Delta_p^{EN \rightarrow D_0}]$	$[\delta_p^{EN \rightarrow D_1}, \Delta_p^{EN \rightarrow D_1}]$
Q_0	$[\delta_p^{Q_0 \rightarrow D_0}, \Delta_p^{Q_0 \rightarrow D_0}]$	$[\delta_p^{Q_0 \rightarrow D_1}, \Delta_p^{Q_0 \rightarrow D_1}]$
Q_1	$[\delta_p^{Q_1 \rightarrow D_0}, \Delta_p^{Q_1 \rightarrow D_0}]$	$[\delta_p^{Q_1 \rightarrow D_1}, \Delta_p^{Q_1 \rightarrow D_1}]$

D_0 : is a function of EN, Q_0, Q_1

D_1 : is a function of EN, Q_0, Q_1

How long does it take for D_0 to change from EN
min. delay & max. delay

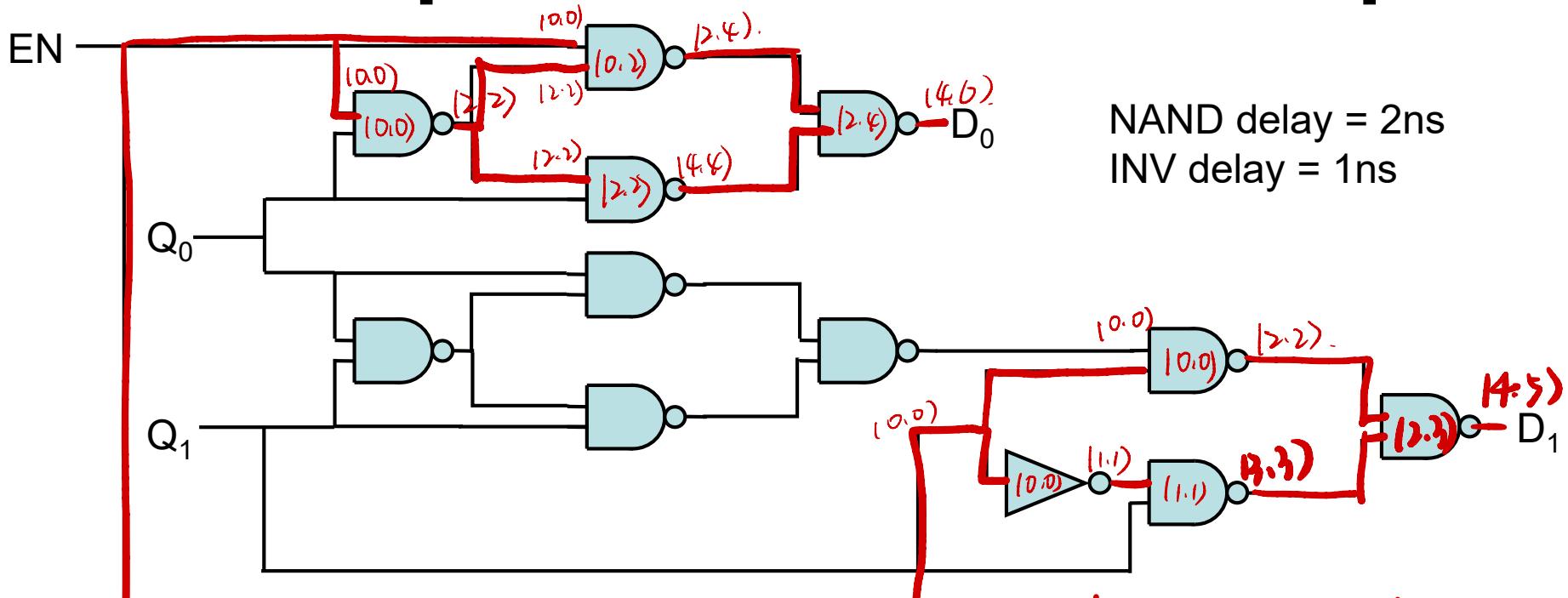
Gate Implementation of Example



	D_0	D_1
EN		
Q_0	(14.6)	(18.10)
Q_1		

	D_0	D_1
EN	$[\delta_p^{EN \rightarrow D_0}, \Delta_p^{EN \rightarrow D_0}]$	$[\delta_p^{EN \rightarrow D_1}, \Delta_p^{EN \rightarrow D_1}]$
Q_0	$[\delta_p^{Q_0 \rightarrow D_0}, \Delta_p^{Q_0 \rightarrow D_0}]$	$[\delta_p^{Q_0 \rightarrow D_1}, \Delta_p^{Q_0 \rightarrow D_1}]$
Q_1	$[\delta_p^{Q_1 \rightarrow D_0}, \Delta_p^{Q_1 \rightarrow D_0}]$	$[\delta_p^{Q_1 \rightarrow D_1}, \Delta_p^{Q_1 \rightarrow D_1}]$

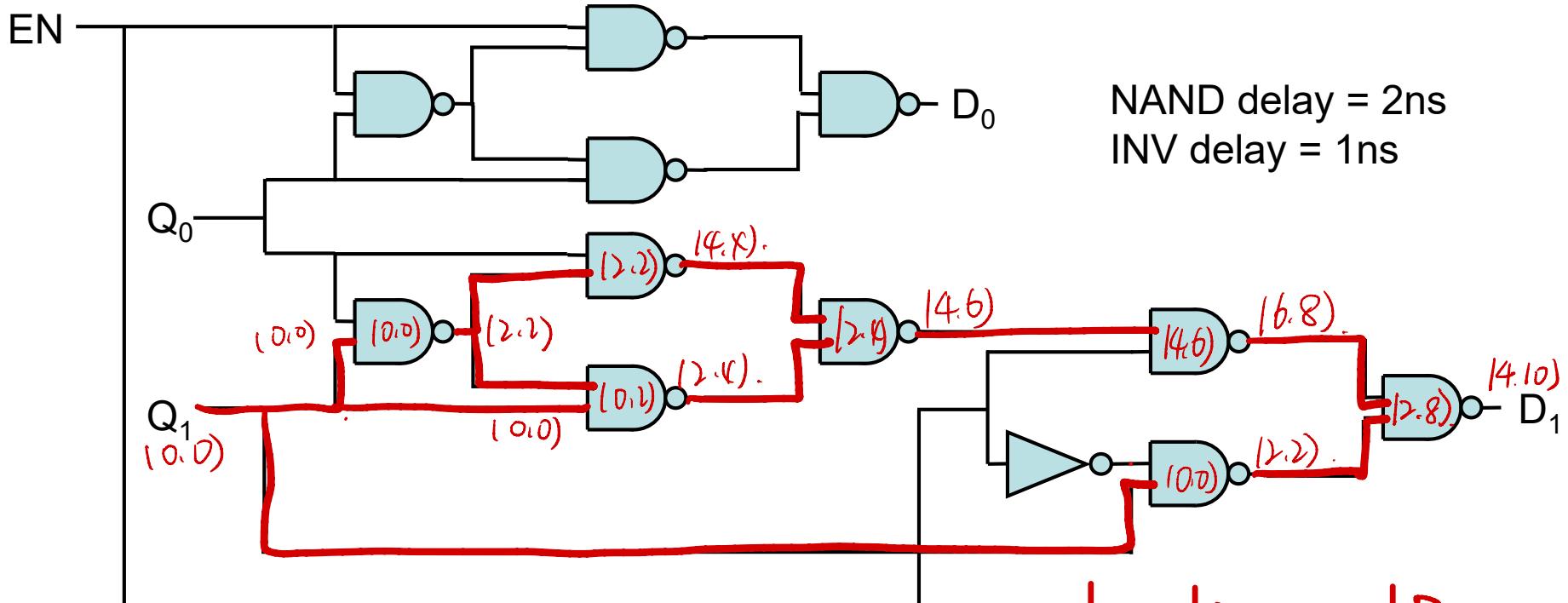
Gate Implementation of Example



	D_0	D_1
EN	(14.6)	(14.5)
Q_0		
Q_1		

	D_0	D_1
EN	$[\delta_p^{EN \rightarrow D_0}, \Delta_p^{EN \rightarrow D_0}]$	$[\delta_p^{EN \rightarrow D_1}, \Delta_p^{EN \rightarrow D_1}]$
Q_0	$[\delta_p^{Q_0 \rightarrow D_0}, \Delta_p^{Q_0 \rightarrow D_0}]$	$[\delta_p^{Q_0 \rightarrow D_1}, \Delta_p^{Q_0 \rightarrow D_1}]$
Q_1	$[\delta_p^{Q_1 \rightarrow D_0}, \Delta_p^{Q_1 \rightarrow D_0}]$	$[\delta_p^{Q_1 \rightarrow D_1}, \Delta_p^{Q_1 \rightarrow D_1}]$

Gate Implementation of Example



	D_0	D_1
EN	$[\delta_p^{EN \rightarrow D_0}, \Delta_p^{EN \rightarrow D_0}]$	$[\delta_p^{EN \rightarrow D_1}, \Delta_p^{EN \rightarrow D_1}]$
Q_0	$[\delta_p^{Q_0 \rightarrow D_0}, \Delta_p^{Q_0 \rightarrow D_0}]$	$[\delta_p^{Q_0 \rightarrow D_1}, \Delta_p^{Q_0 \rightarrow D_1}]$
Q_1	$[\delta_p^{Q_1 \rightarrow D_0}, \Delta_p^{Q_1 \rightarrow D_0}]$	$[\delta_p^{Q_1 \rightarrow D_1}, \Delta_p^{Q_1 \rightarrow D_1}]$
D_0		
D_1		
EN		
Q_0		
Q_1	$(\infty, -\infty)$	(4.10)

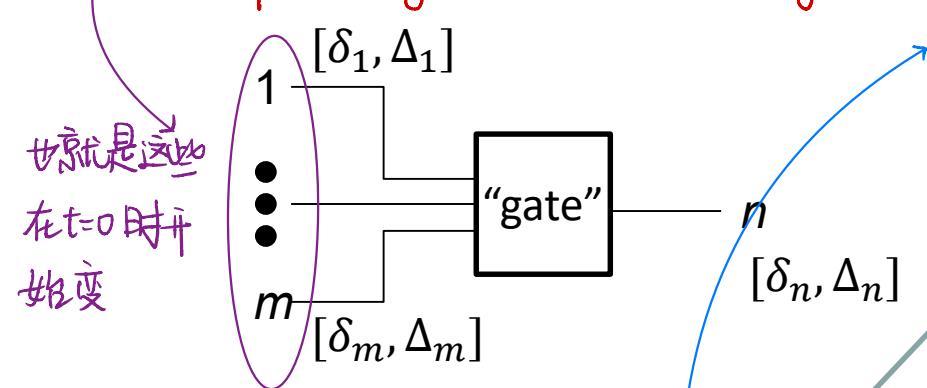
Path Delay Equations

$$[\delta_i, \Delta_i] = \begin{cases} [0, 0] & \text{if } i \text{ is a primary input/FF output} \\ \text{min and max delay to } i & \text{if } i \text{ is a "gate" output} \end{cases}$$

Wire 直接连接
要通过 gates

We assume the primary input & state output change at time zero in the cycle.

So the earliest input will determine when the output start to change. & the latest input will determine when the output stop to change (ignore the logic) worst case.



$$\delta_n = \min_{i \in [1, m]} (\delta_i + t_p^{i \rightarrow n})$$

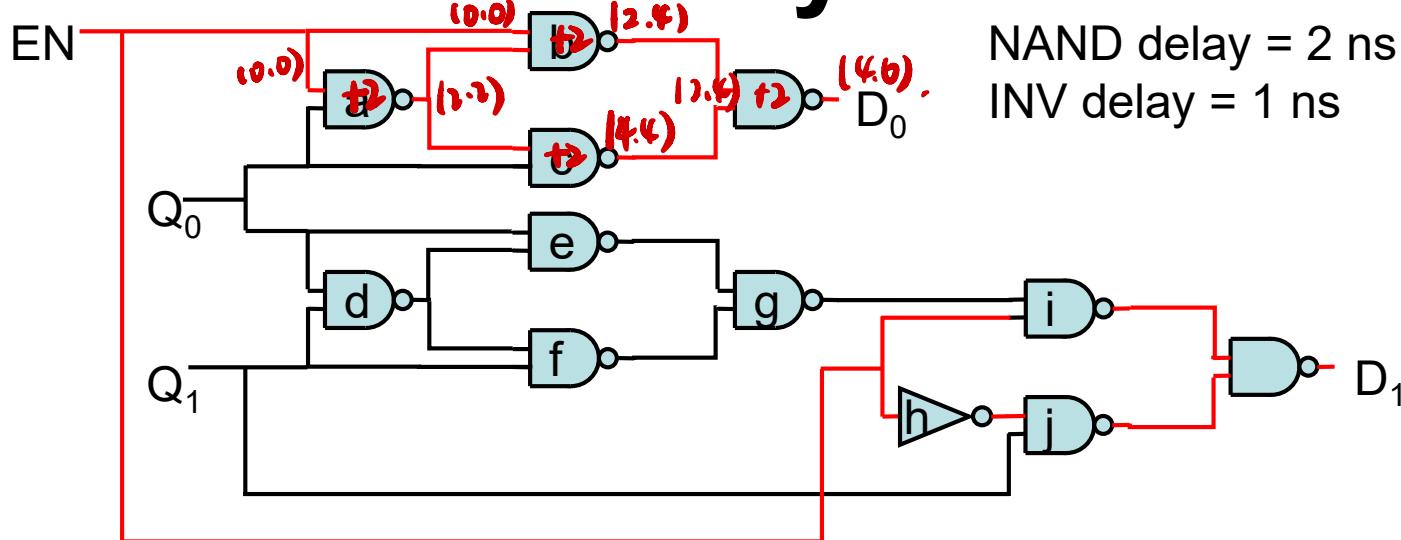
$$\delta_n = t_p^{\text{"gate"}} + \min_{i \in [1, m]} (\delta_i) \quad \leftarrow \text{start to change}$$

$$\Delta_n = \max_{i \in [1, m]} (\Delta_i + t_p^{i \rightarrow n})$$

$$\Delta_n = t_p^{\text{"gate"}} + \max_{i \in [1, m]} (\Delta_i) \quad \leftarrow \text{stop to change}$$

Assuming equal delay from all gate inputs

Min & Max Delays From EN



$$[\delta_{EN}, \Delta_{EN}] = [0, 0]$$

$$[\delta_a, \Delta_a] = 2 + [\min(\delta_{EN}), \max(\Delta_{EN})] = 2 + [0, 0] = [2, 2]$$

$$[\delta_b, \Delta_b] = 2 + [\min(\delta_{EN}, \delta_a), \max(\Delta_{EN}, \Delta_a)] = 2 + [\min(0, 2), \max(0, 2)] = 2 + [0, 2] = [2, 4]$$

$$[\delta_c, \Delta_c] = 2 + [\min(\delta_a), \max(\Delta_a)] = 2 + [2, 2] = [4, 4]$$

$$[\delta_{D_0}, \Delta_{D_0}] = 2 + [\min(\delta_b, \delta_c), \max(\Delta_b, \Delta_c)] = 2 + [\min(2, 4), \max(4, 4)] = 2 + [2, 4] = [4, 6]$$

$$[\delta_h, \Delta_h] = 1 + [\min(\delta_{EN}), \max(\Delta_{EN})] = 1 + [0, 0] = [1, 1]$$

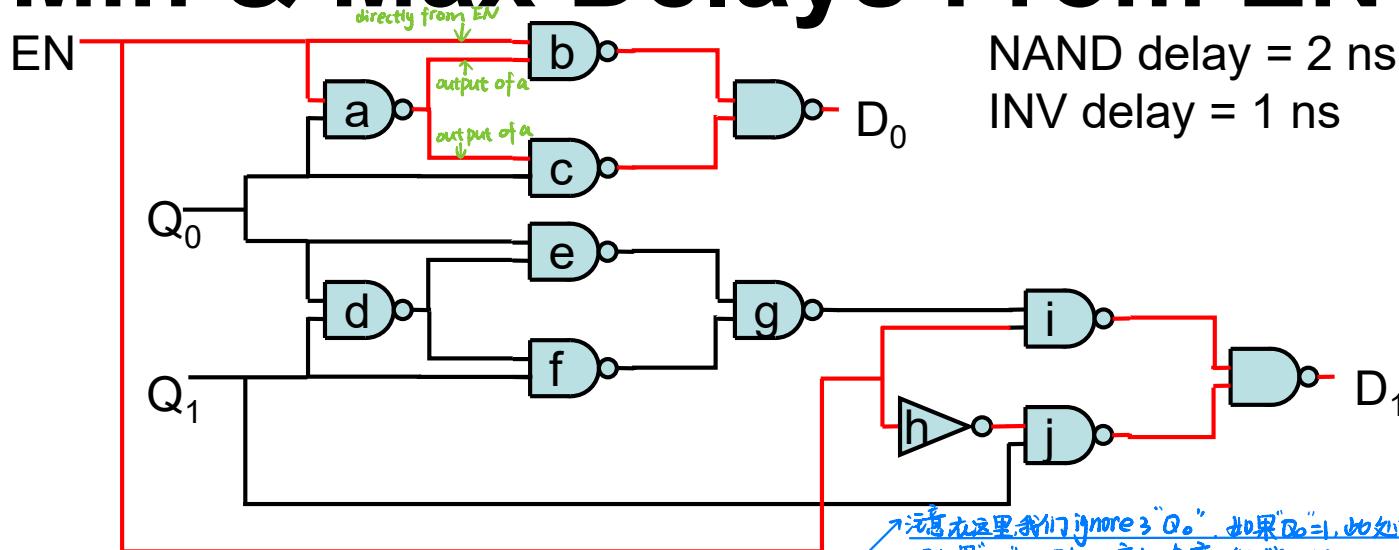
$$[\delta_i, \Delta_i] = 2 + [\min(\delta_{EN}), \max(\Delta_{EN})] = 2 + [0, 0] = [2, 2]$$

$$[\delta_j, \Delta_j] = 2 + [\min(\delta_h), \max(\Delta_h)] = 2 + [1, 1] = [3, 3]$$

$$[\delta_{D_1}, \Delta_{D_1}] = 2 + [\min(\delta_i, \delta_j), \max(\Delta_i, \Delta_j)] = 2 + [\min(2, 3), \max(2, 3)] = 2 + [2, 3] = [4, 5]$$

Red lines: All the path that EN will influence D₀ and D₁

Min & Max Delays From EN



$$EN: [\delta_{EN}, \Delta_{EN}] = [0, 0] \quad EN \text{ changes at } t=0. \text{ (make a clean transition)}$$

$$a: [\delta_a, \Delta_a] = 2 + [\min(\delta_{EN}), \max(\Delta_{EN})] = 2 + [0, 0] = [2, 2]$$

the output at "a" will change : NAND gate 的 delay to L

$$b: [\delta_b, \Delta_b] = 2 + [\min(\delta_{EN}, \delta_a), \max(\Delta_{EN}, \Delta_a)] = 2 + [\min(0, 2), \max(0, 2)] = 2 + [0, 2] = [2, 4]$$

EN 变成从 min delay 到 max delay. 因为 EN 就变一次从 0->1 或 1->0. so $\delta = \Delta$

$$c: [\delta_c, \Delta_c] = 2 + [\min(\delta_a), \max(\Delta_a)] = 2 + [2, 2] = [4, 4]$$

$$D_0: [\delta_{D_0}, \Delta_{D_0}] = 2 + [\min(\delta_b, \delta_c), \max(\Delta_b, \Delta_c)] = 2 + [\min(2, 4), \max(4, 4)] = 2 + [2, 4] = [4, 6]$$

a_t A_t for the late

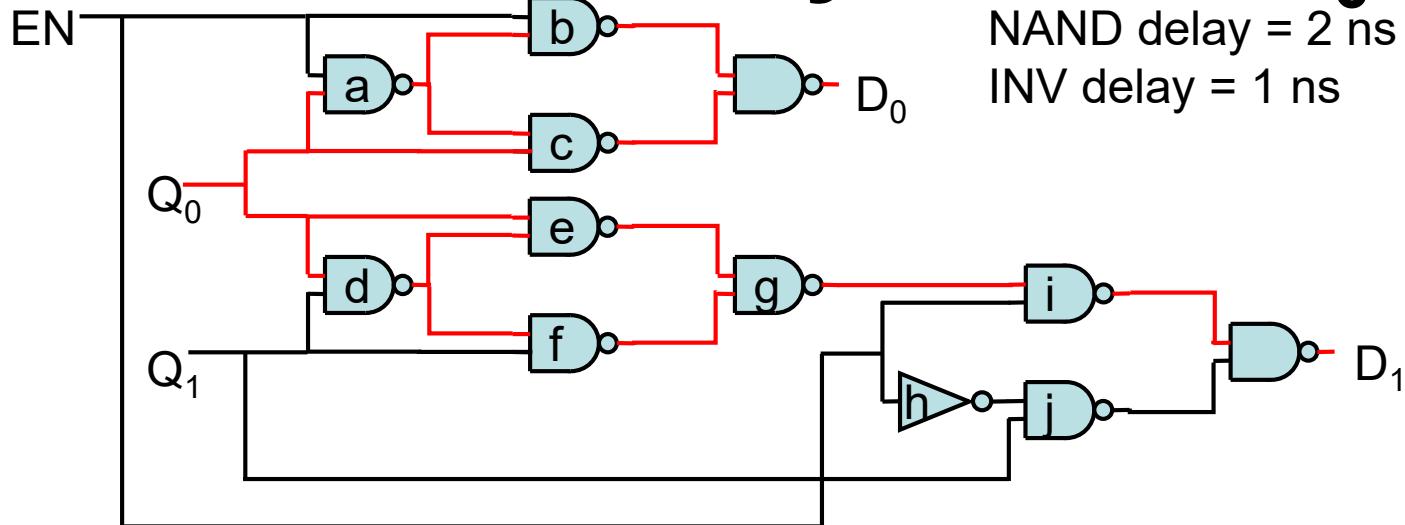
$$[\delta_h, \Delta_h] = 1 + [\min(\delta_{EN}), \max(\Delta_{EN})] = 1 + [0, 0] = [1, 1]$$

$$[\delta_i, \Delta_i] = 2 + [\min(\delta_{EN}), \max(\Delta_{EN})] = 2 + [0, 0] = [2, 2]$$

$$[\delta_j, \Delta_j] = 2 + [\min(\delta_h), \max(\Delta_h)] = 2 + [1, 1] = [3, 3]$$

$$[\delta_{D_1}, \Delta_{D_1}] = 2 + [\min(\delta_i, \delta_j), \max(\Delta_i, \Delta_j)] = 2 + [\min(2, 3), \max(2, 3)] = 2 + [2, 3] = [4, 5]$$

Min & Max Delays From Q₀



$$[\delta_{Q_0}, \Delta_{Q_0}] = [0, 0]$$

$$[\delta_a, \Delta_a] = 2 + [\min(\delta_{Q_0}), \max(\Delta_{Q_0})] = 2 + [0, 0] = [2, 2]$$

$$[\delta_b, \Delta_b] = 2 + [\min(\delta_a), \max(\Delta_a)] = 2 + [2, 2] = [4, 4]$$

$$[\delta_c, \Delta_c] = 2 + [\min(\delta_{Q_0}, \delta_a), \max(\Delta_{Q_0}, \Delta_a)] = 2 + [\min(0, 2), \max(0, 2)] = 2 + [0, 2] = [2, 4]$$

$$[\delta_{D_0}, \Delta_{D_0}] = 2 + [\min(\delta_b, \delta_c), \max(\Delta_b, \Delta_c)] = 2 + [\min(4, 2), \max(4, 4)] = 2 + [2, 4] = [4, 6]$$

$$[\delta_d, \Delta_d] = 2 + [\min(\delta_{Q_0}), \max(\Delta_{Q_0})] = 2 + [0, 0] = [2, 2]$$

$$[\delta_e, \Delta_e] = 2 + [\min(\delta_{Q_0}, \delta_d), \max(\Delta_{Q_0}, \Delta_d)] = 2 + [\min(0, 2), \max(0, 2)] = 2 + [0, 2] = [2, 4]$$

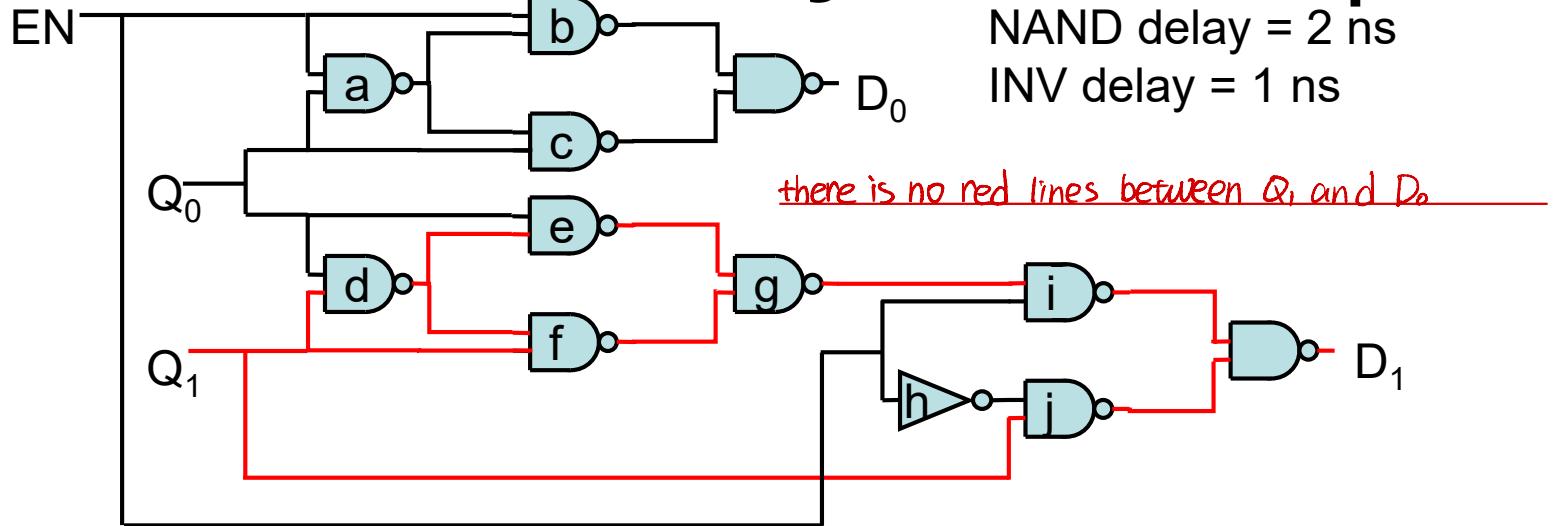
$$[\delta_f, \Delta_f] = 2 + [\min(\delta_d), \max(\Delta_d)] = 2 + [2, 2] = [4, 4]$$

$$[\delta_g, \Delta_g] = 2 + [\min(\delta_e, \delta_f), \max(\Delta_e, \Delta_f)] = 2 + [\min(2, 4), \max(4, 4)] = 2 + [2, 4] = [4, 6]$$

$$[\delta_i, \Delta_i] = 2 + [\min(\delta_g), \max(\Delta_g)] = 2 + [4, 6] = [6, 8]$$

$$[\delta_{D_1}, \Delta_{D_1}] = 2 + [\min(\delta_i), \max(\Delta_i)] = 2 + [6, 8] = [8, 10]$$

Min & Max Delays From Q_1



$$[\delta_{Q_1}, \Delta_{Q_1}] = [0, 0]$$

$$[\delta_d, \Delta_d] = 2 + [\min(\delta_{Q_1}), \max(\Delta_{Q_1})] = 2 + [0, 0] = [2, 2]$$

$$[\delta_e, \Delta_e] = 2 + [\min(\delta_d), \max(\Delta_d)] = 2 + [2, 2] = [4, 4]$$

$$[\delta_f, \Delta_f] = 2 + [\min(\delta_{Q_1}, \delta_d), \max(\Delta_{Q_1}, \Delta_d)] = 2 + [\min(0, 2), \max(0, 2)] = 2 + [0, 2] = [2, 4]$$

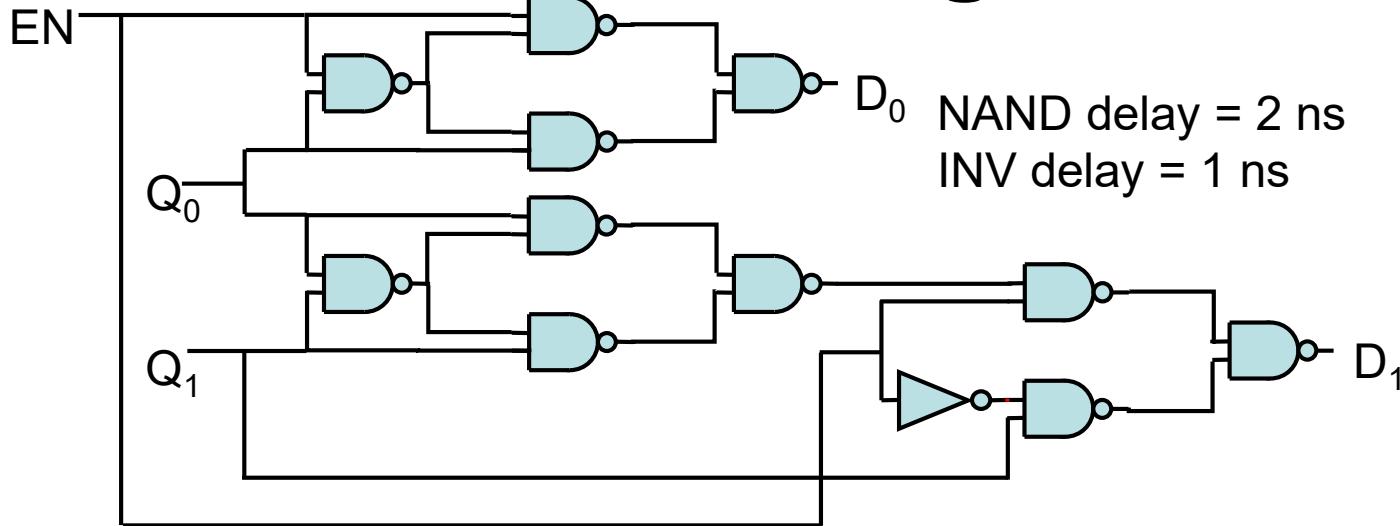
$$[\delta_g, \Delta_g] = 2 + [\min(\delta_e, \delta_f), \max(\Delta_e, \Delta_f)] = 2 + [\min(4, 2), \max(4, 4)] = 2 + [2, 4] = [4, 6]$$

$$[\delta_i, \Delta_i] = 2 + [\min(\delta_g), \max(\Delta_g)] = 2 + [4, 6] = [6, 8]$$

$$[\delta_j, \Delta_j] = 2 + [\min(\delta_{Q_1}), \max(\Delta_{Q_1})] = 2 + [0, 0] = [2, 2]$$

$$[\delta_{D_1}, \Delta_{D_1}] = 2 + [\min(\delta_i, \delta_j), \max(\Delta_i, \Delta_j)] = 2 + [\min(6, 2), \max(8, 2)] = 2 + [2, 8] = [4, 10]$$

Combinational Timing Summary



From EN

$$[\delta_p^{EN \rightarrow D_0}, \Delta_p^{EN \rightarrow D_0}] = [4, 6]$$

$$[\delta_p^{EN \rightarrow D_1}, \Delta_p^{EN \rightarrow D_1}] = [4, 5]$$

From Q₀

$$[\delta_p^{Q_0 \rightarrow D_0}, \Delta_p^{Q_0 \rightarrow D_0}] = [4, 6]$$

$$[\delta_p^{Q_0 \rightarrow D_1}, \Delta_p^{Q_0 \rightarrow D_1}] = [8, 10]$$

From Q₁

$$[\delta_p^{Q_1 \rightarrow D_1}, \Delta_p^{Q_1 \rightarrow D_1}] = [4, 10]$$

D₀ is unaffected by Q₁

不是 D delay 而是 No delay

	D ₀	D ₁
EN	[4,6]	[4,5]
Q ₀	[4,6]	[8,10]
Q ₁	[∞, -∞]	[4,10]

因为我们要在EN, Q₀, Q₁之间做比较

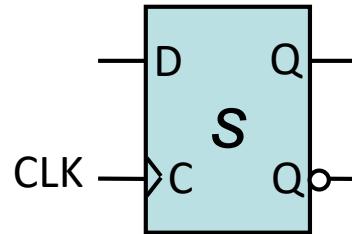
而又他们中每个人 min 中的最小值

再入他们中每个人 max 中的最大值

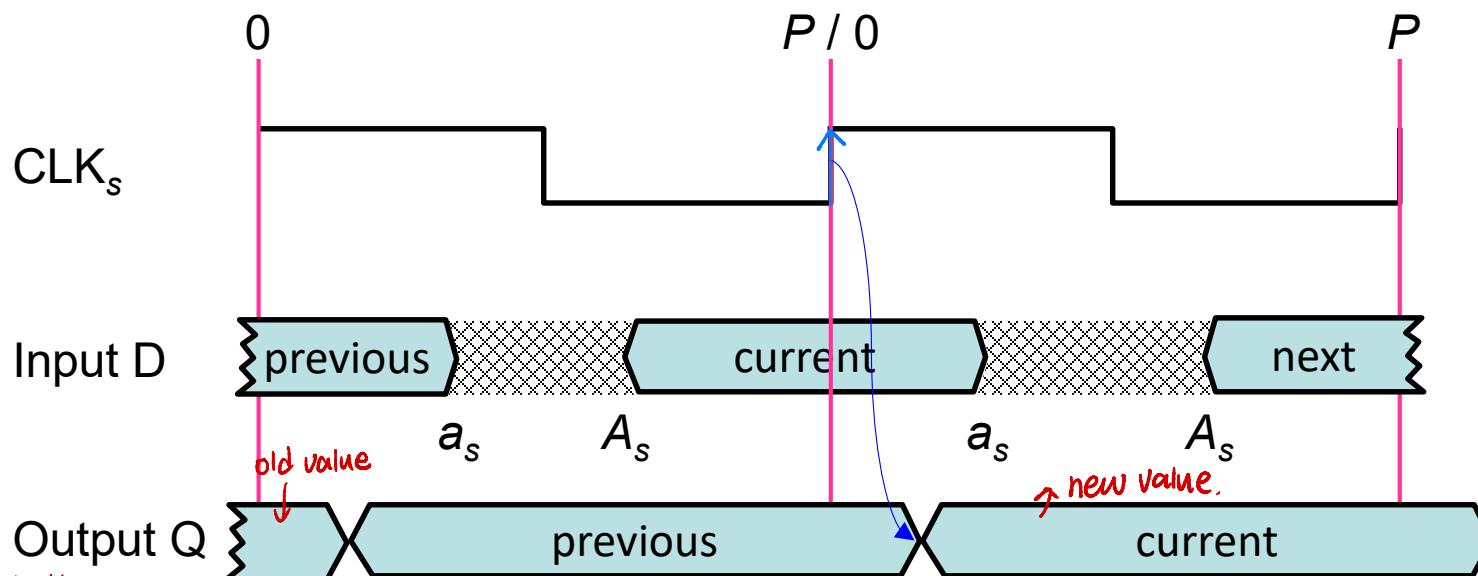
Signal “Departure” Times

Q: when does it start actually propagating its new value?
We call it departure time.

Positive Edge-Triggered D Flip-Flop



Now, this destination register now has become a source register or source flip-flops
For other flip flops



flip flop 離開時間：

即使 input D 不斷變動 (即變 stable 之前). 但因為是 edge triggering.
所以它們有 clean transition on the state.

one change per cycle.

conclusion : Input might have multiple transitions, but the state makes a single transition because edge triggering

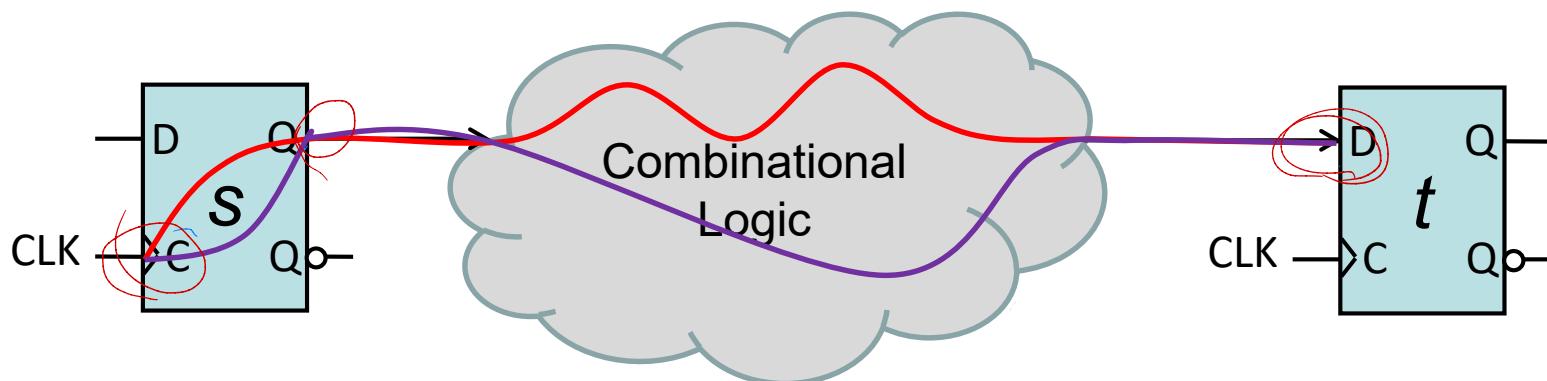
$$d_s = D_s = t_p = \text{departure time}$$

$d_s = D_s = t_p = \frac{\text{delay within the latch}}{\text{propagation}}$

t_p clock \uparrow $C_s \rightarrow Q_s$ \uparrow Q_{output}

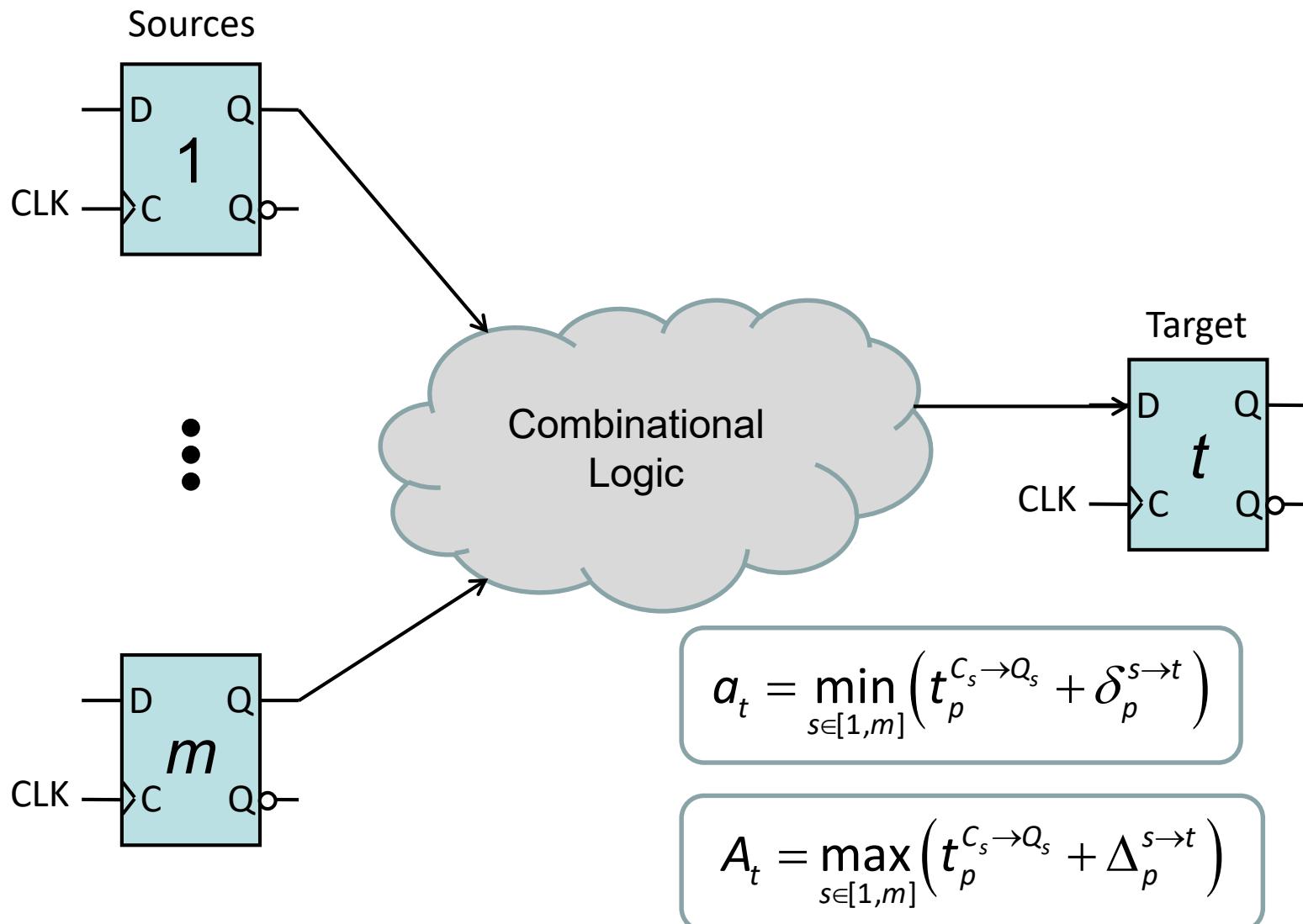
As soon as Q becomes stable, then D becomes stable at this time.

Signal Propagation Model

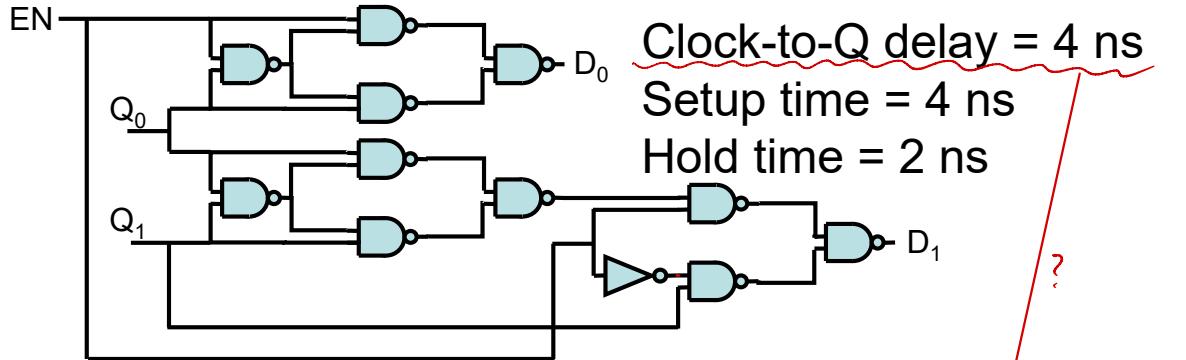


	Shortest path	Longest path
Departure time from s	$d_s = t_p^{C_s \rightarrow Q_s}$	$D_s = t_p^{C_s \rightarrow Q_s}$
Arrival time at t	$a_t = d_s + \delta_p^{s \rightarrow t}$	$A_t = D_s + \Delta_p^{s \rightarrow t}$
	$= t_p^{C_s \rightarrow Q_s} + \delta_p^{s \rightarrow t}$	$= t_p^{C_s \rightarrow Q_s} + \Delta_p^{s \rightarrow t}$

Signal Propagation Equations



Detailed Timing Analysis



Departure times (from FFs): $d_0 = D_0 = d_1 = D_1 = t_p^{C \rightarrow Q} = 4 \text{ ns}$

ignore the input, because input is not synchronized with the clock.

	D_0	D_1
EN	[4,6]	[4,5]
Q_0	[4,6]	[8,10]
Q_1	[∞ , ∞]	[4,10]

But the state is synchronized with the clock.

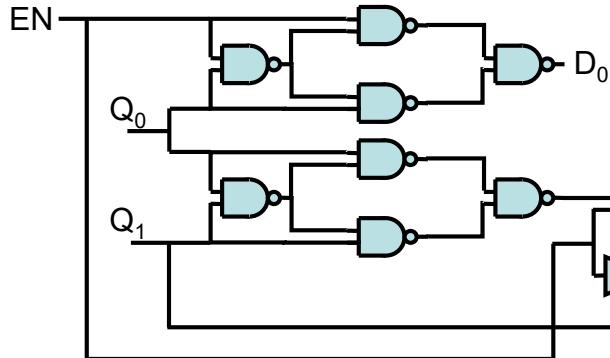
arrival time at flip flop 0
 $a_0 = \min(t_p^{C \rightarrow Q} + \delta_p^{Q_0 \rightarrow D_0}, t_p^{C \rightarrow Q} + \delta_p^{Q_1 \rightarrow D_0})$
 $(\stackrel{D_0}{=})$
 $= \min(4+4, 4+\infty)$
 $= 8 \text{ ns}$

$$A_0 = \max(t_p^{C \rightarrow Q} + \Delta_p^{Q_0 \rightarrow D_0}, t_p^{C \rightarrow Q} + \Delta_p^{Q_1 \rightarrow D_0})$$
 $= \max(4+6, 4+\infty)$
 $= 10 \text{ ns}$

$$a_1 = \min(t_p^{C \rightarrow Q} + \delta_p^{Q_0 \rightarrow D_1}, t_p^{C \rightarrow Q} + \delta_p^{Q_1 \rightarrow D_1})$$
 $= \min(4+8, 4+4)$
 $= 8 \text{ ns}$

$$A_1 = \max(t_p^{C \rightarrow Q} + \Delta_p^{Q_0 \rightarrow D_1}, t_p^{C \rightarrow Q} + \Delta_p^{Q_1 \rightarrow D_1})$$
 $= \max(4+10, 4+10)$
 $= 14 \text{ ns}$

Detailed Timing Analysis (Cont'd)



Clock-to-Q delay = 4 ns
Setup time S = 4 ns
Hold time H = 2 ns

	D_0	D_1
EN	[4,6]	[4,5]
Q_0	[4,6]	[8,10]
Q_1	[∞,∞]	[4,10]

$$a_0 = 8 \text{ ns}$$

$$a_1 = 8 \text{ ns}$$

Hold Requirement

$$a \geq H$$

$$a_0 \geq H \quad \checkmark$$

$$a_1 \geq H \quad \checkmark$$

$$A_0 = 10 \text{ ns}$$

$$A_1 = 14 \text{ ns}$$

Setup Requirement

$$A \leq P - S \Rightarrow P \geq A + S$$

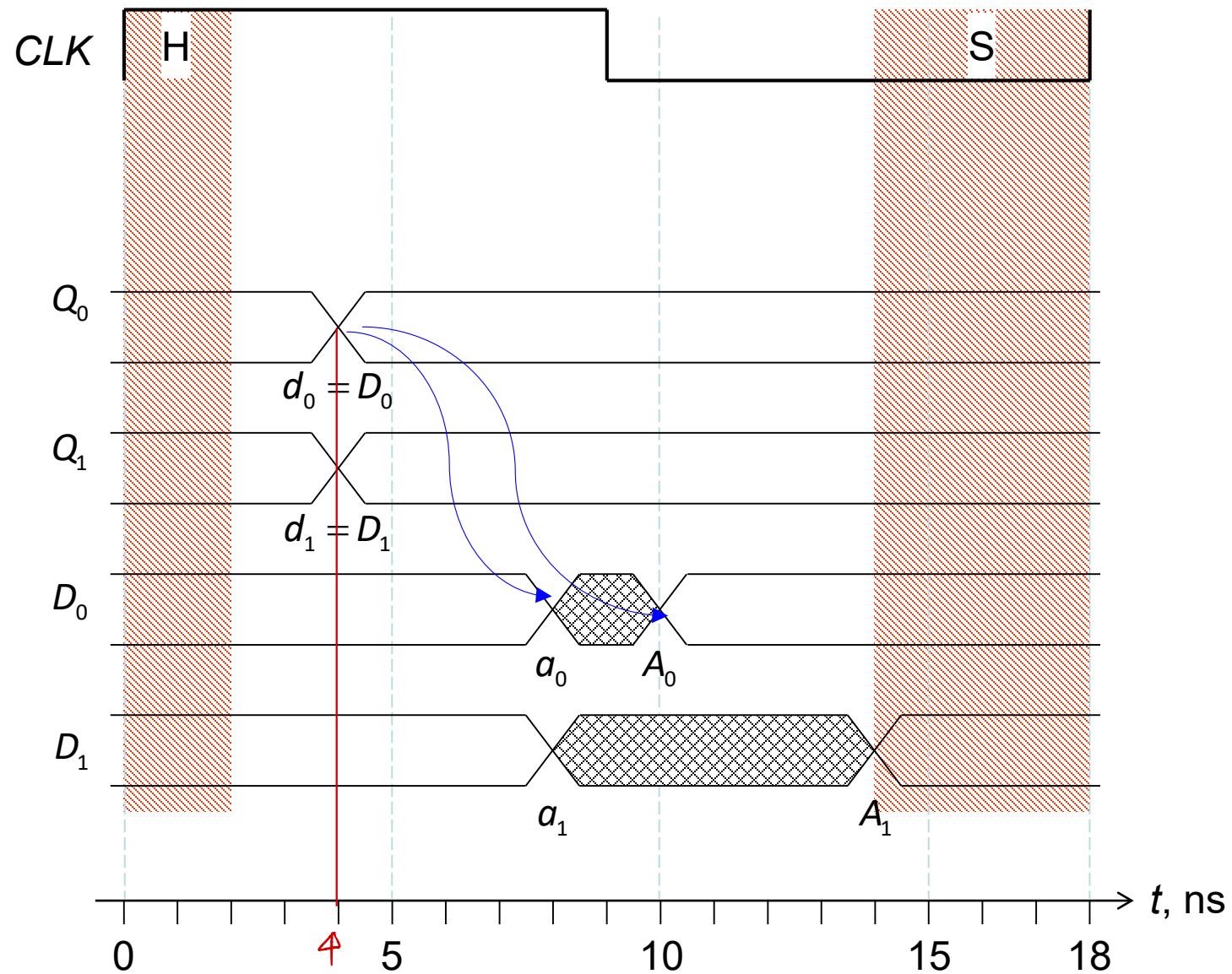
$$P \geq A_0 + S = 10 + 4 = 14 \text{ ns}$$

$$P \geq A_1 + S = 14 + 4 = 18 \text{ ns}$$

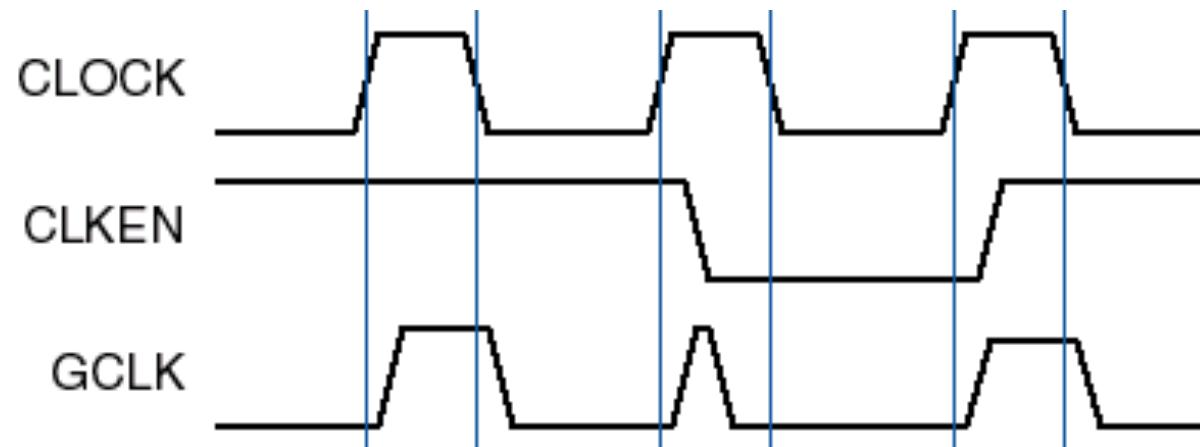
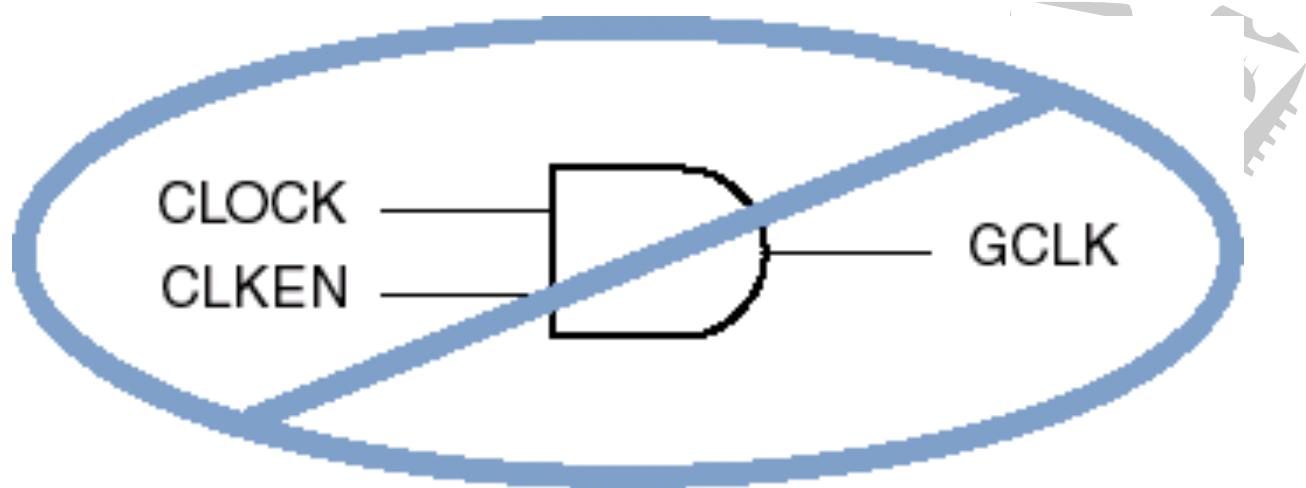
∴ Hold requirements are met.

$$\therefore P_{\min} = 18 \text{ ns}$$

Timing Waveforms



Gating the clock



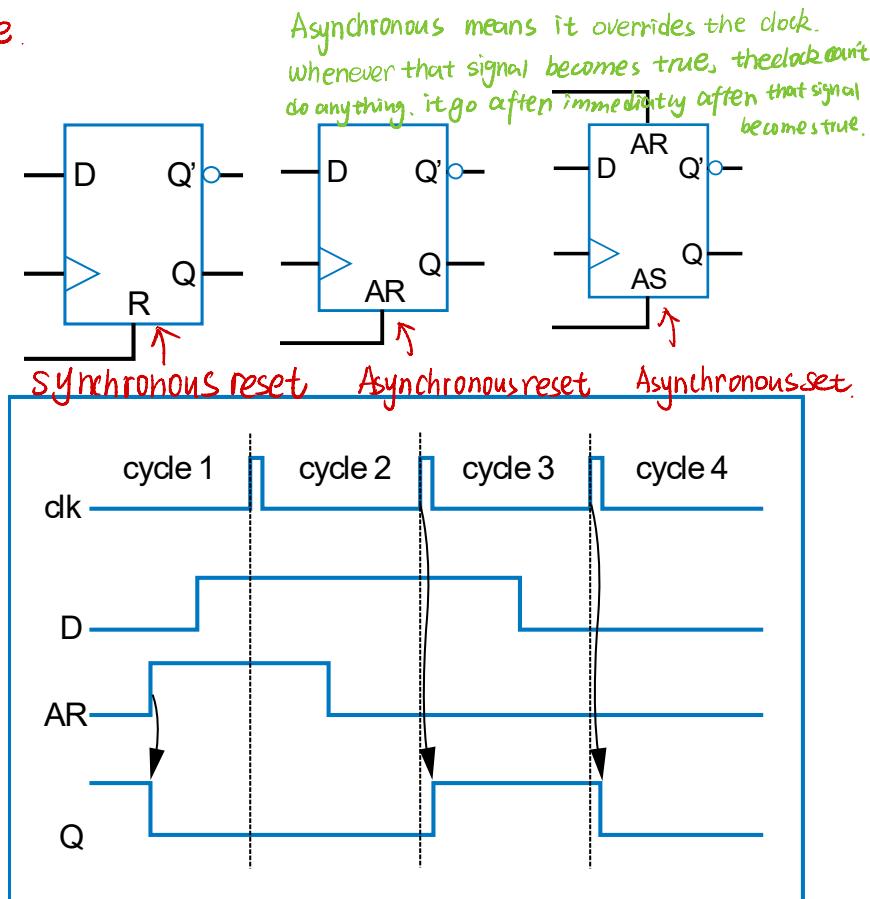
- Definitely a no-no
 - Glitches possible if control signal (CLKEN) is generated by the same clock
 - Excessive clock skew in any case.

Initialization

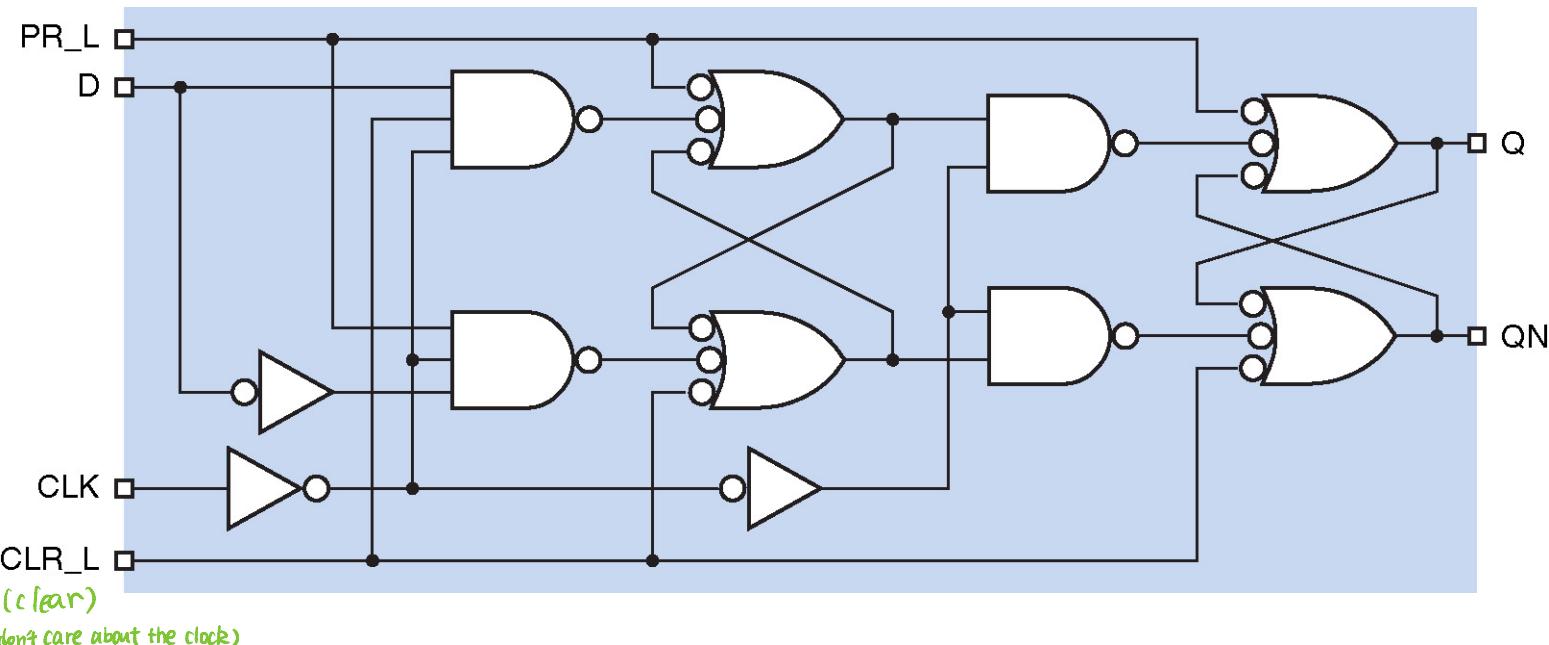
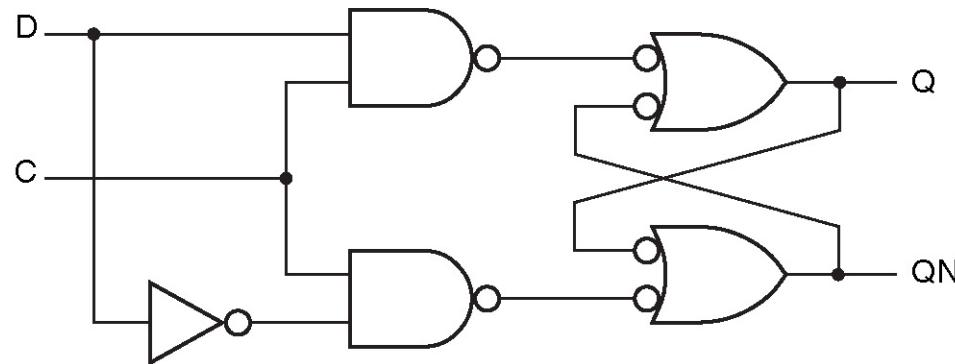
Synchronous vs. Asynchronous Set/Reset

reset means make it zero, set means make it one.

- Synchronous reset: clears Q to 0 on next clock edge
- Synchronous set: sets Q to 1 on next clock edge
- Asynchronous reset: clear Q to 0 immediately (not dependent on clock edge)
 - Example timing diagram shown
- Asynchronous set: set Q to 1 immediately



Implementation of Asynchronous Set/Reset



Asynchronous inputs*



anything that is not controlled by the clock that is running the circuit.

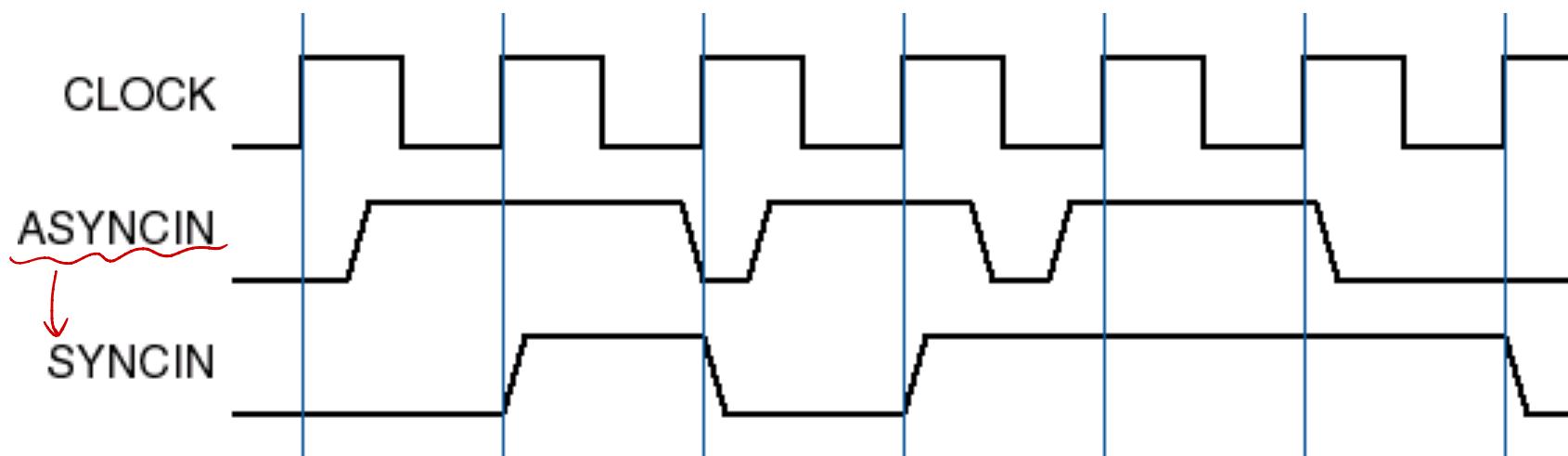
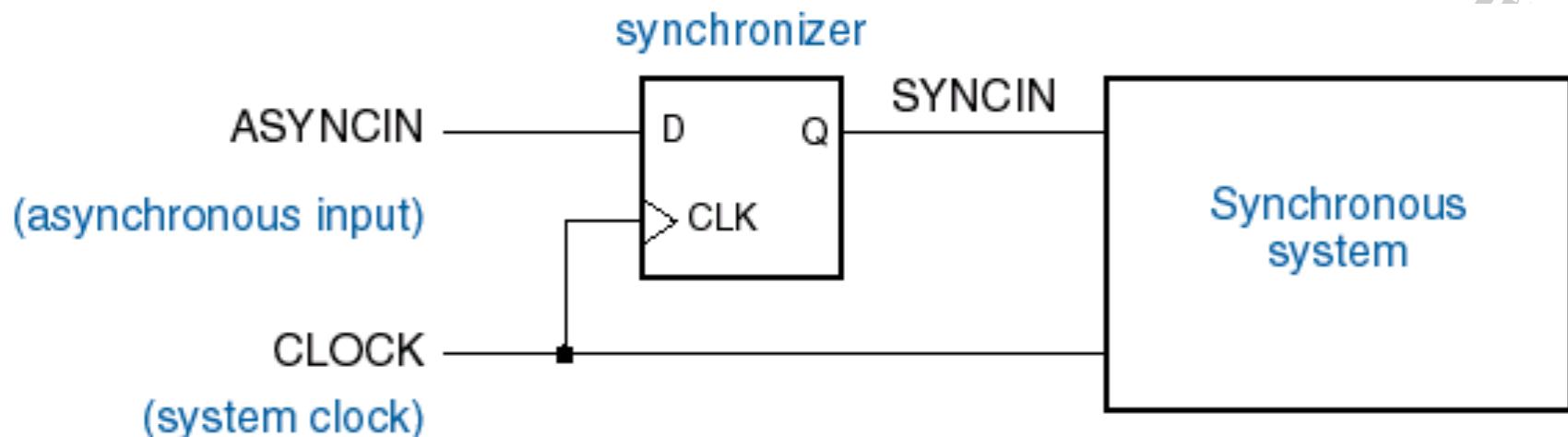
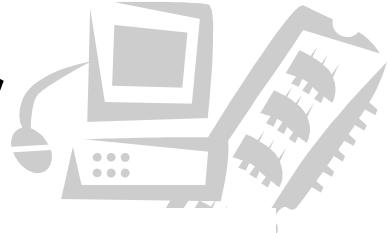
→ it is a source of metastability. 因为不知道什么时候改变。

- Not all inputs are synchronized with the clock
- Examples:
 - Keystrokes
 - Sensor inputs
 - Data received from a network (transmitter has its own clock)
这些 AI change 的 frequency 一定比 clock 慢。
但我们不知道他具体啥时候变，他们也许会在高 clock-edge 非常薄弱的地方改变。→造成 havoc 在 circuit 中。
- Inputs must be synchronized with the system clock before being applied to a synchronous system.

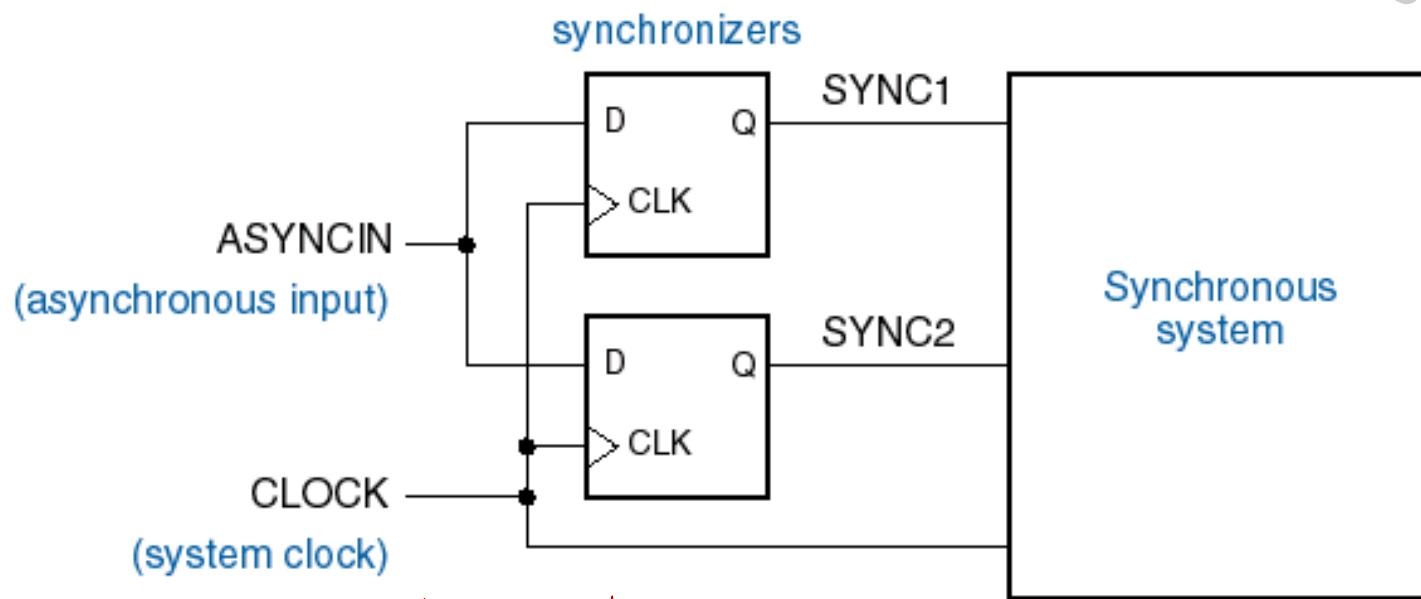
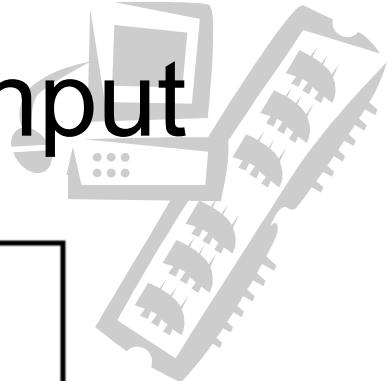
next page
解决方法:

Task ①: synchronize it..

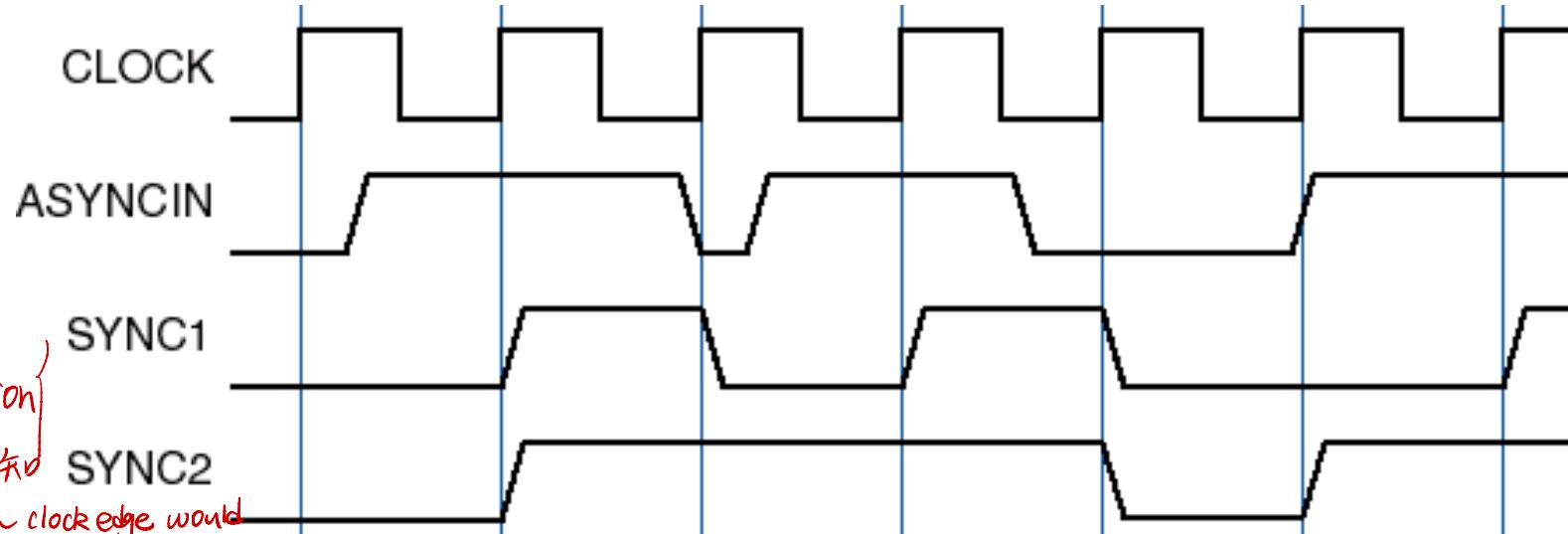
A simple synchronizer



Only one synchronizer per input

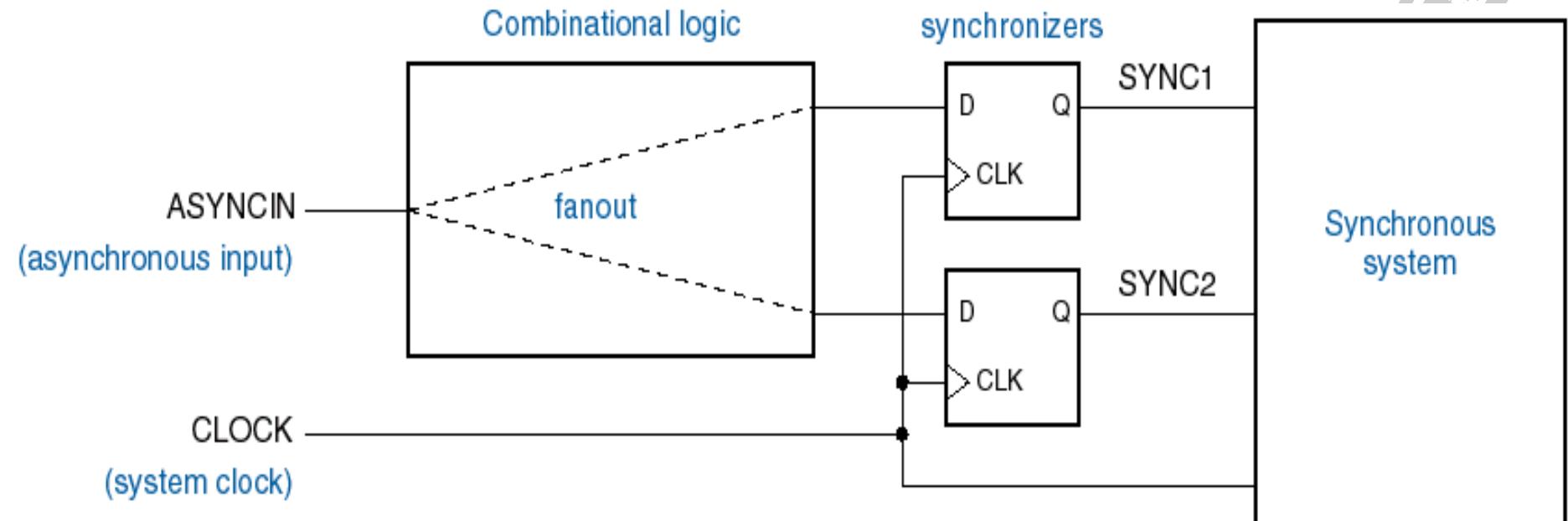


Wait for my input, catch it and move on





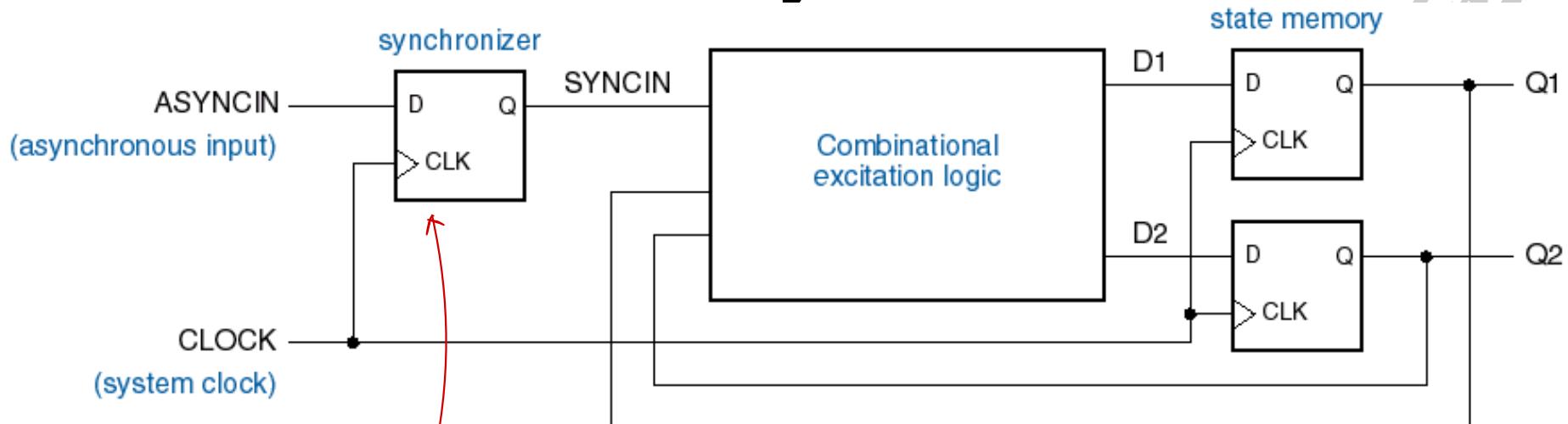
Even worse



- Combinational delays to the two synchronizers are likely to be different.

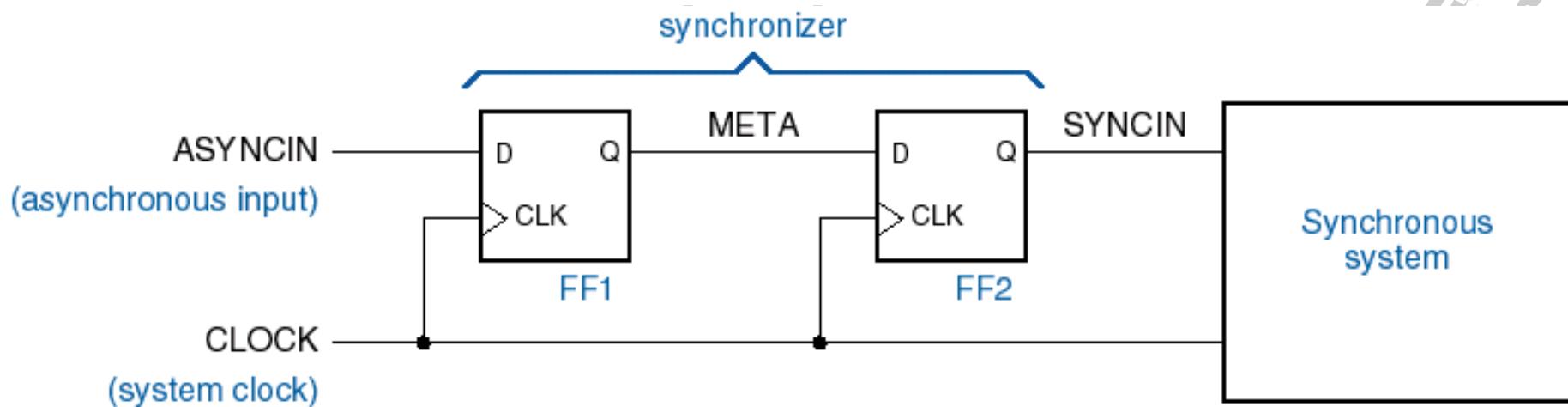


The way to do it



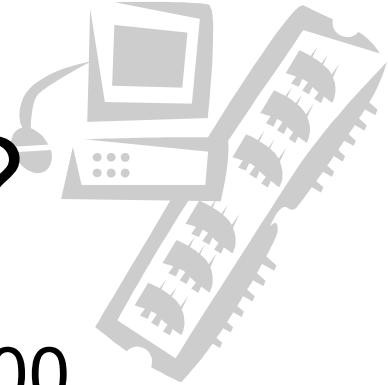
- One synchronizer per input
- Carefully locate the synchronization points in a system.
- But still a problem -- the synchronizer output may become metastable when setup and hold time are not met.

Recommended synchronizer



- Hope that FF1 settles down before “META” is sampled.
 - In this case, “SYNCIN” is valid for almost a full clock period.
 - Can calculate the probability of “synchronizer failure” (FF1 still metastable when META sampled)
 - MTBF: Mean Time Between Failures

Is 1000 years enough?



- If MTBF = 1000 years and you ship 52,000 copies of the product, then some system experiences a mysterious failure every week.
- Real-world MTBFs must be much higher.
- How to get better MTBFs?
 - Use faster flip-flops
 - But clock speeds keep getting faster, thwarting this approach.
 - Wait for multiple clock ticks to get a longer metastability resolution time
 - Waiting longer usually doesn't hurt performance
 - ...unless there is a critical "round-trip" handshake.

